# Computer Networks

## Exercise Session 09

Prof. Dr. Oliver Hahm

Frankfurt University of Applied Sciences
Faculty 2: Computer Science and Engineering
oliver.hahm@fb2.fra-uas.de
https://teaching.dahahm.de

December 23, 2022

# General Schedule

All exercises will follow this general schedule

- Identify potential understanding problems
  → Ask your questions
  → Recap of the lecture
- Address the understanding problems
  → Answer your questions
  → Repeat certain topics
- Walk through the exercises/solutions → Some hints and guidance
  → Work time or presentation of results

# Inter-Networking

You have seen ...

- how different networks are connected via a router
- which mechanisms are involved when forwarding a packet to a different network
- what an AS is
- the difference between routing and forwarding

# Network Layer: Routing Schemes

You have seen . . .

- the requirements for a routing protocol
- how routing algorithms can be categorized
- flooding and hot-potato as examples for local routing algorithms
- the difference between source routing and hop-by-hop routing
- the difference between reactive and proactive routing algorithms
- how metrics are used to calculate the path costs

# Network Layer: Distance Vector Routing

You have seen . . .

- that distance vector routing protocols exchange forwarding tables between neighbors
- RIP as an example for a distance vector routing protocol
- how the Bellman-Ford Algorithm works
- what the Count-to-Infinity problem is
- how Split Horizon (with Poison Reversed) can be used to mitigate this problem

# Network Layer: Link State Routing

You have seen . . .

- that link state routing protocols exchange information between all routers
- OSPF as an example for a link state routing protocol
- that OSPF allows for routing hierarchies
- how the Dijkstra Algorithm works

# Network Layer: More Routing Protocols

You have seen . . .

- IS-IS as another example for a link state routing protocol
- RPL as routing protocol for resource-constrained node networks (aka IOT networks)
- OLSR as link state routing protocol for wireless ad-hoc networks
- BGP as an example for an inter-domain routing protocol

# IPv4: Calculate checksum

**RFC 791, page 14**

*„The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero“.*

- To calculate the checksum of the packet, the sum of each 2 byte word inside the header must be calculated. The checksum field itself is skipped here!
  `4500 + 0034 + B612 + 4000 + 4006 + 0A00 + 008B + 5BC6 + AEE0 = 2907D`
- Next, the result of the calculation is converted to binary:
  `2907D` $\Longrightarrow$ `10 1001 0000 0111 1101`
- The first two bits are the carry and need to be added to the rest of the value:
  `10 + 1001 0000 0111 1101 = 1001 0000 0111 1111`
- Next, every bit of the result is flipped to obtain the checksum:
  `1001 0000 0111 1111`
  `=> 0110 1111 1000 0000`
- The result `0110 1111 1000 0000` is equal to the value `6F80` in hexadecimal notation, as already shown in the original IP packet header.

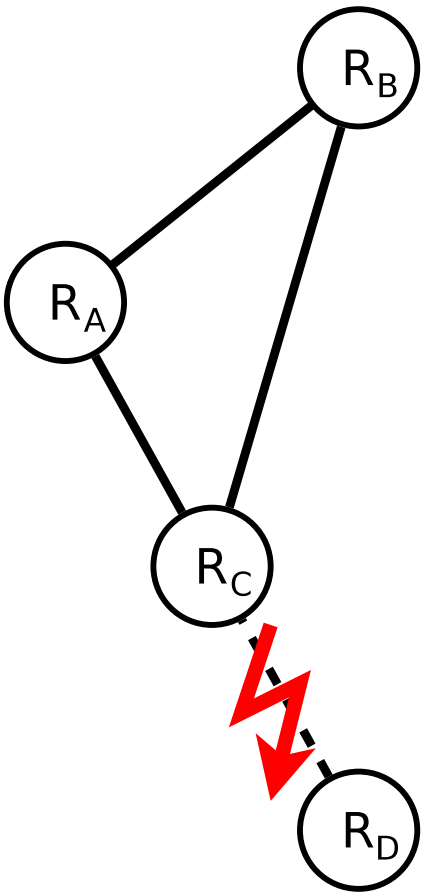
# IPv4: Verify checksum

- To verify a checksum, the same procedure is used as above, with a single exception: The original header checksum is not omitted.
  `4500 + 0034 + B612 + 4000 + 4006 + 6F80 + 0A00 + 008B + 5BC6 + AEE0 = 2FFFD`
- Next, the result of the calculation is converted to binary:
  `2FFFD` $\Longrightarrow$ `10 1111 1111 1111 1101`
- The first two bits are the carry and need to be added to the rest of the value:
  `10 + 1111 1111 1111 1101 = 1111 1111 1111 1111`
- Next, every bit of the result is flipped:
  `1111 1111 1111 1111`
  `=> 0000 0000 0000 0000`
- This indicates: No error detected! Any result, which is $\neq$ `0` indicates: Error!

*Source: RFC 791 and Wikipedia*

# Example where Split Horizon fails

- If the connection between $R_C$ and $R_D$ fails, $R_C$ labels $R_D$ in his local routing table as not accessible

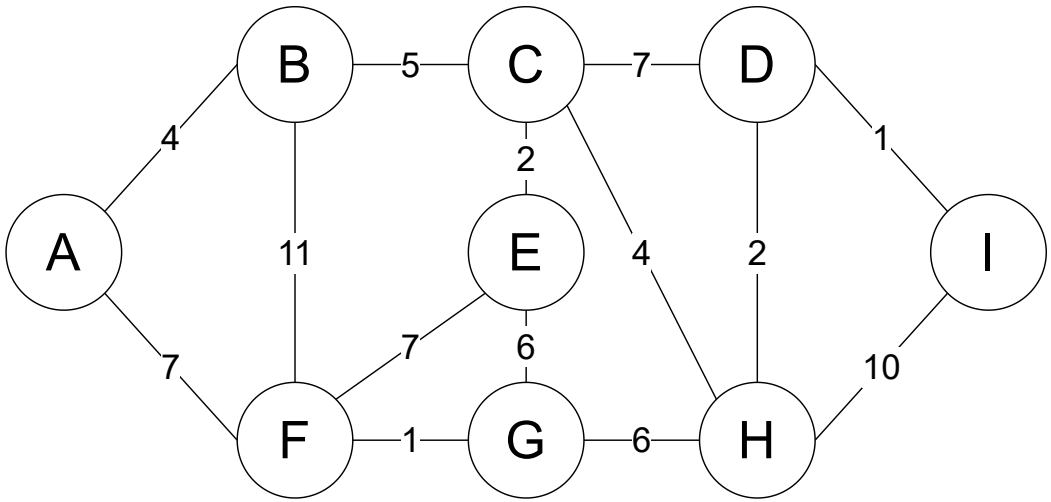



- $R_C$ informs $R_A$ and $R_B$, that $R_D$ cannot be reached
- If the advertisement message arrives first at $R_A$, it assumes the best route to $R_D$ is via $R_B$
- $R_A$ informs $R_B$ that $R_D$ cannot be reached and informs $R_C$ that it reaches $R_D$ by 3 hops
- $R_C$ believes that it can reach $R_D$ via $R_A$ by 4 hops and it informs $R_B$ that it has a route to $R_D$
- $R_B$ informs $R_A$ that it reaches $R_D$ by 5 hops
- $R_A$ informs $R_C$ that it reaches $R_D$ by 6 hops
- ...

$\Longrightarrow$ **Count-to-Infinity**

# Dijkstra Algorithm

- Given: The following network

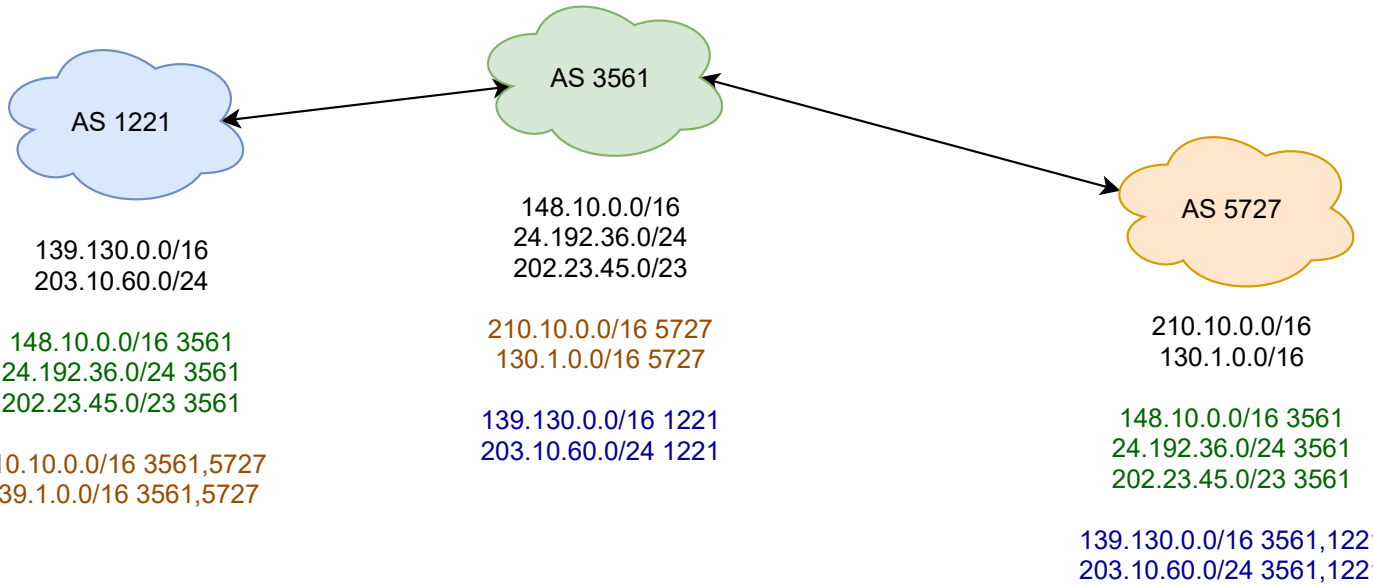



1. Determine the spanning tree of shortest paths using the link state routing protocol (Dijkstra's algorithm) of node A.

Source: Jörg Roth. **Prüfungstrainer Rechnernetze: Aufgaben und Lösungen**. Vieweg (2010)

# BGP

- BGP router discover AS paths to the prefixes of their neighbors → their BGP peers
- Theses paths are stored in a table: the BGP RIB
  - This may include redundant paths
- BGP defines a default-free zone ⇒ all participating routers must provide a complete RIB ⇒ no default route
- Example:



# BGP Selection Criteria

BGP router have to select paths and make forwarding decisions:

1. Preference
   - Assign all RIB entries a preference - based on local policies and attributes
2. Route Selection
   - For each prefix the router selects based on the following criteria:
     1. Highest preference
     2. Shortest path
     3. Additional tie breaker rules
   - Selected routes are put into the FIB
3. Route Forwarding
   - Before the propagation of the FIB entries these are filtered according to the policies